

Pixellaser: Learning Range via Texture

Michael Leece, Nicole Lesperance, Steve Matsumoto,
Max Korbel, Kenny Lei, and Zachary Dodds

Harvey Mudd College Computer Science
301 Platt Boulevard
Claremont, CA 91711

mleece@hmc.edu, lesper@cooper.edu, smatsumoto@hmc.edu, mkorbel@hmc.edu, dodds@cs.hmc.edu (contact)

Abstract

The problem of finding a robot's range-to-obstacles is a fundamental one with an elegant solution: the laser range finder (LRF). This work has developed algorithms for replacing a laser with a camera for indoor applications. Our approach uses machine learning algorithms to segment the groundplane from single images flexibly, quickly, and robustly. We then transform those segmentations into laserscan-like estimates of local conditions. Current work is investigating whether off-the-shelf algorithms for mapping, localization, and navigation with LRFs work without alteration using these "pixel"-scans.

Motivation

Most autonomous platforms use sensors that directly compute range from time-of-flight, e.g. from laser range finders (LRFs). Yet future generations of commodity platforms – descendants of the service and entertainment robots now in homes -- are not likely to use LRFs. Monocular images offer an advantageous alternative to LRFs along several axes: cameras are less power-hungry, less heavy, less bulky, less range-limited, and, perhaps most importantly, less expensive.

Usual range-from-vision approaches use temporal feature correspondence across a monocular image stream to deduce distance from pixels (Kanade et al. 2001). More recently work has sought to "fill in the gaps" for featureless areas by learning distance directly from image texture. This project builds from Plagemann et al. (2008): here, we use larger image patches and ordinary webcams instead of an omniscam and single-pixel columns as features. Preliminary results underscore the power of range-from-texture approaches, pioneered in Horswill's Polly (Horswill 1995), and used in many systems since.

Approach

Figure 1 summarizes our approach: we use a nearest-neighbors patch classifier, based on both texture and color

parameters, in order to classify small patches of image as either groundplane or obstacles, presuming an indoor environment.

The resulting segmentations yield "pixelscans" that can be rendered in a top-down coordinate system and compared to ground-truth. In order to focus on our current results, we refer to other sources for the details of this approach (Lesperance et al. 2010).

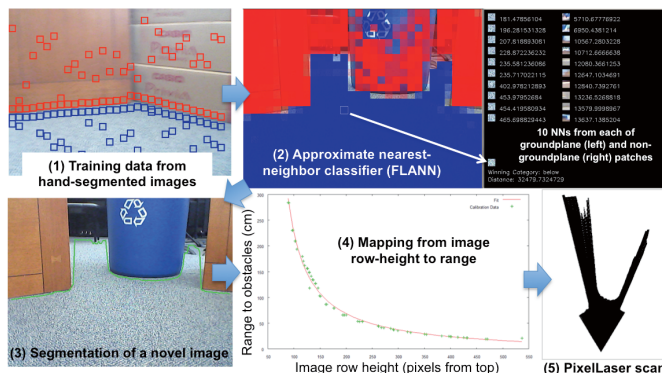


Fig 1. The Pixellaser processing pipeline

Results

Each stage of the above pipeline has been evaluated across several different data sets. Figure 2 shows our platform, a netbook mounted on top of an iRobot Create. No sensing other than the netbook's built-in webcam was used. Two cropped images from our "lobby" environment are also shown.

As Figure 3 attests, the classification accuracy for distinguishing "obstacle" image patches from "traversable" patches is quite good. The accuracy is over 90% for several distinct environments and sets of conditions. Note, too, that the number of nearest neighbors used to determine the classification (the color bars) does not have a significant effect on accuracy.

From those strong results, we use a bottom-up search for the strongest transition between traversable and untraversable texture. The strongest transition is "snapped" to a close, strong intensity edge when one is present. We used several specific strategies including multi-resolution

search, different sizes of vertical spatial context, and even a third nearest-neighbor classification tree that sought to recognize the “edge” patches that included the boundary between traversable and untraversable space. Of the best of these strategies, all demonstrated similar qualitative performance, as suggested in Figure 4: the median absolute pixel error is very near zero, but misclassifications do lead to spikes in the errors reported by the system.



Figure 2. The Create platform and netbook (right) and two images, taken from the netbook, within one of the four environments in which we tested PixelLaser’s pipeline (right)

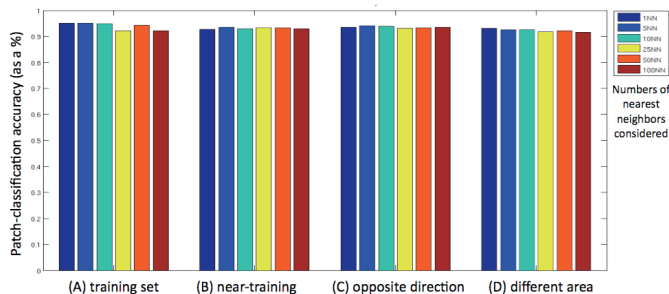


Figure 3. The accuracy of the nearest-neighbors classifier in determining whether image patches are groundplane or not. Note that the results are quite good regardless of environment or number of nearest neighbors used.

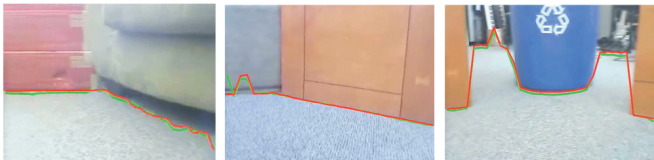


Figure 4. Three examples of groundplane segmentations resulting from the classifications shown in Figure 3. Image patches at a horizontal resolution of 20 pixels provide the raw segmentation shown in green; the red line has “snapped” to nearby strong edges, when present.

Applications

We tested these scans with several off-the-shelf algorithms that ordinarily take laser scans as input. For example, Figure 5 presents snapshots of both training images and an extended autonomous run at AAI 2010’s Robotics Exhibition Education Track. With only PixelLaser sensing, the Create circumnavigated the exhibition hall four times without incident over a 15-minute span. Figure 6 shows a CoreSLAM-generated map using scans from our test environment at Harvey Mudd College.

These results suggest that monocular vision can, indeed, provide *quantitative*, as well as qualitative, range-to-obstacle scans. Those scans promise to make off-the-shelf spatial reasoning algorithms accessible to a much broader set of robot platforms than could use them up to now.

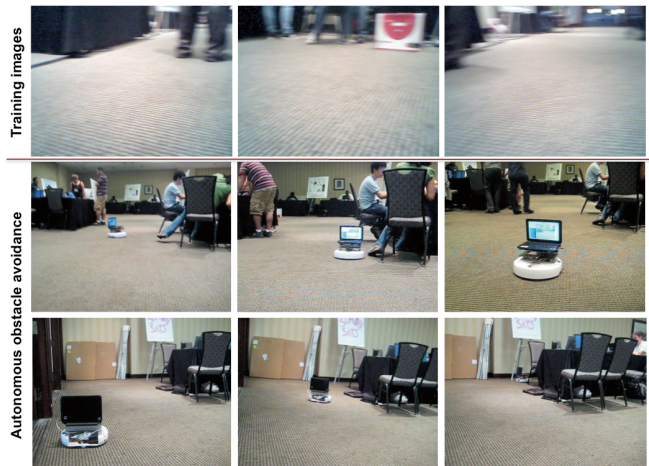


Figure 5. (Top) training images from a completely distinct environment (AAAI 2010’s venue); below are snapshots from an extended autonomous run in which the robot used no sensing other than the PixelLaser scans to avoid obstacles.

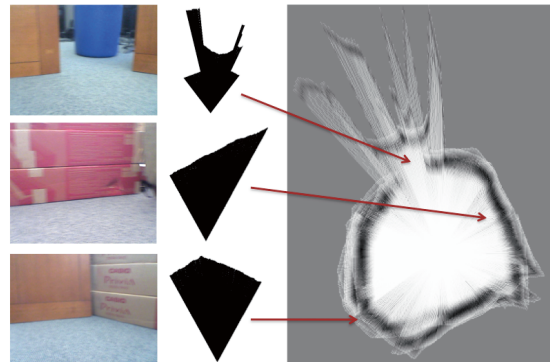


Figure 6. Images, scans, and the resulting map from Figure 4’s environment. The CoreSLAM algorithm integrated the scans, unchanged from its original implementation for laser data.

References

- Horswill, Ian. (1995) Analysis of Adaptation and Environment. *Artificial Intelligence*, Vol. 73, pp. 1-30. 1995.
- Kanade, T., Kanade, B. Y., and Morris, D. D. (2001) Factorization methods for structure from motion. *Phil. Trans. of the Royal Society of London, Series A*. 356, pp.1153-1173, 2001.
- Lesperance, N., Leece, M., Matsumoto, S., Korbelt, M., Lei, K., and Dodds, Z. (2010) PixelLaser: Computing Range from Monocular Texture. *Proceedings, International Symposium on Visual Computing (ISVC '10)* Nov. 29 – Dec. 1, 2010.
- Plagemann, C., Enres, F., Hess, J., Stachniss, C., and Burgard, W. (2008) Monocular Range Sensing: A non-parametric learning approach. *Proceedings, ICRA 2008* May 19-23, pp. 929-934.